Problem A
FizzBuzz
Problem ID: fizzbuzz
Time Limit: 1 second

According to Wikipedia, FizzBuzz is a group word game for children to teach them about division. This may or may not be true, but this question is generally used to ~~torture~~ screen young computer science graduates during programming interviews.

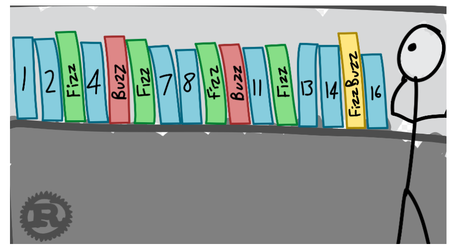

Image by chris morgan cc by

Basically, this is how it works: you print the integers from 1 to N, replacing any of them divisible by X with `Fizz` or, if they are divisible by Y, with `Buzz`. If the number is divisible by both X and Y, you print `FizzBuzz` instead.

Check the samples for further clarification.

Input

Input file will contain a single test case. Each test case will contain three integers on a single line, X, Y and N ($1 \leq X < Y \leq N \leq 100$).

Output

Print integers from 1 to N in order, each on its own line, replacing the ones divisible by X with `Fizz`, the ones divisible by Y with `Buzz` and ones divisible by both X and Y with `FizzBuzz`.

Sample Input	Sample Output
2 3 7	1 Fizz Buzz Fizz 5 FizzBuzz 7

Sample Input	Sample Output
2 4 7	1 Fizz 3 FizzBuzz 5 Fizz 7

Sample Input	Sample Output
3 5 7	1 2 Fizz 4 Buzz Fizz 7

Problem B
Election
Problem ID: election
Time Limit: 1 second

After all the fundraising, campaigning and debating, the election day has finally arrived. Only two candidates remain on the ballot and you work as an aide to one of them.

Image by democratic alliance cc by-sa 2.0

Reports from the voting stations are starting to trickle in and you hope that you can soon declare a victory.

There are N voters and everyone votes for one of the two candidates (there are no spoiled ballots). In order to win, a candidate needs more than half of the votes. A certain number $M \leq N$ of ballots have been counted, and there are V_i votes for candidate i ($V_1 + V_2 = M$), where V_1 is the number of votes your candidate received.

Due to the historical data and results of highly scientific polls, you know that each of the remaining votes has a 50% chance to go to your candidate. That makes you think that you could announce the win before all the votes are counted. So, if the probability of winning strictly exceeds a certain threshold W, the victory is yours! We just hope you are sure of this, we don't want any scandals...

Input

The first line of input contains a single positive integer $T \leq 100$ indicating the number of test cases. Next T lines each contain four integers: N, V_1, V_2 and W as described above.

The input limits are as follows:
$1 \leq N \leq 50$
$50 \leq W < 100$
$V_1, V_2 \geq 0$
$V_1 + V_2 \leq N$

Output

For each test case print a single line containing the appropriate action:

- If the probability that your candidate will win is strictly greater than $W\%$, print

  `GET A CRATE OF CHAMPAGNE FROM THE BASEMENT!`

- If your candidate has no chance of winning, print

  `RECOUNT!`

- Otherwise, print

  `PATIENCE, EVERYONE!`

Sample Input	Sample Output
4 5 0 3 75 5 0 2 75 6 1 0 50 7 4 0 75	RECOUNT! PATIENCE, EVERYONE! PATIENCE, EVERYONE! GET A CRATE OF CHAMPAGNE FROM THE BASEMENT!

Problem C
Password Hacking
Problem ID: password
Time Limit: 1 second

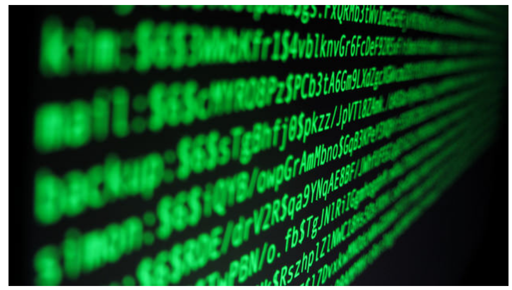

Photo by christiaan colen cc by-sa 2.0

We all know that passwords are not very secure unless users are disciplined enough to use passwords that are difficult to guess. But most users are not so careful, and happily use passwords such as "123456". In fact, there are lists of commonly used passwords that hackers can use for breaking into systems, and these passwords often work.

You have done a lot of hacking using such lists, and you have a good idea of how likely each password in the list is the correct one (you are very surprised by the number of people using "123456" as their passwords). You have a new account to hack, and you have decided to try each of the passwords in the list one at a time, until the correct one is found. You are absolutely sure that the account you want to hack uses a password in the given list.

What is the expected number of attempts to find the correct passwords, assuming that you try these passwords in the optimal order?

Input

The first line of input contains a positive integer N, the number of passwords in the list. Each of the next N lines gives the password, followed by a space, followed by the probability that the password is the correct one. Each password is a non-empty string consisting only of alphanumeric characters and is up to 12 characters long. Each probability is a real number with 4 decimal places. You may assume that there are at most 500 passwords in the list, and that the sum of all probabilities equals 1. No two passwords in the list are the same.

Output

Output on a single line the expected number of attempts to find the correct passwords using the optimal order. Answers within 10^{-4} of the correct answer will be accepted.

Sample Input	Sample Output
2 123456 0.6666 qwerty 0.3334	1.3334

Sample Input	Sample Output
3 qwerty 0.5432 123456 0.3334 password 0.1234	1.5802

Problem D
Delivering Goods
Problem ID: delivering
Time Limit: 5 seconds

Photo by kamyar adl cc by-sa 2.0

You run a delivery company and must deploy a fleet of vehicles to deliver goods to clients. All of the goods and delivery trucks are initially located at your warehouse.

The road network consists of one-way streets between junctions. The warehouse and clients are all located at a junction. You know the driving time across each street.

You guarantee extremely fast shipping: the trucks start driving immediately at the start of the day and each client i will receive the package at time T_i where T_i is the shortest possible driving time for a truck to go from the warehouse to the location of the client i.

What is the minimum number of trucks you have to deploy to ensure this guarantee is met? That is, what is the minimum number of trucks such that it is possible to give each truck a driving route so that every client i is visited by some truck at time T_i. Assume it takes no time to load the trucks with the appropriate goods at the start of the day, and it takes no time to drop goods off at a client once the truck arrives at the client. These goods are small enough that each truck can carry goods for as many clients as necessary.

Input

The input consists of a single test case. The first line of each test case consists of three numbers N, M, and C. Here N denotes the number of junctions in the road network ($2 \leq N \leq 10^3$), M denotes the number of streets ($1 \leq M \leq 10^5$), and C denotes the number of clients ($1 \leq C \leq 300$, $C < N$).

The junctions are numbered 0 to $N-1$. The warehouse is always at junction 0. The second line consists of C distinct integers between 1 and $N-1$ indicating the junctions where the clients reside.

The rest of the input consists of M lines, each containing integers U, V, W where $0 \leq U, V \leq N-1$ and $U \neq V$. This indicates there is a one-way street from U to V with driving time W. Each street's driving time W satisfies $1 \leq W \leq 10^9$. It will always be possible to reach every client from the warehouse.

There will be at most one street from a vertex U to another vertex V, but there may be streets from both U to V and from V to U.

Output

Output a single integer that is the minimum number of vehicles required to ensure each client i is visited at time T_i by some vehicle.

Explanations of Sample Inputs

In the first sample, one vehicle can follow the path $(0, 1, 2)$ and the other can follow $(0, 3)$. In the second sample, the only solution is to use paths $(0, 1)$, $(0, 2)$, and $(0, 3)$. In the final sample, one vehicle can follow $(0, 1)$, another $(0, 4, 6)$, and the last one $(0, 2, 3, 5, 7)$.

Sample Input

```
4 5 3
1 2 3
0 1 1
0 3 1
0 2 2
1 2 1
3 2 1
```

Sample Output

```
2
```

Sample Input

```
4 5 3
1 2 3
0 1 1
0 3 1
0 2 1
1 2 1
3 2 1
```

Sample Output

```
3
```

Sample Input

```
8 11 5
1 3 4 6 7
0 1 5
0 4 1
0 2 2
0 6 6
2 3 1
2 6 3
3 5 7
4 1 5
5 7 3
6 5 6
4 6 4
```

Sample Output

```
3
```

Problem E
Studying For Exams
Problem ID: studying
Time Limit: 3 seconds

It is exam time! You have, of course, been spending too much time participating in various programming contests and have not done much studying. Now you have N subjects to study for, but only a limited amount of time before the final exams. You have to decide how much time to allocate to studying each subject, so that your average grade over all N subjects is maximized.

Photo by CollegeDegrees360 cc by-sa 2.0

As a seasoned programming contest competitor, you recognize immediately that you can determine the optimal allocation with a computer program. Of course, you have decided to ignore the amount of time you spend solving this problem (i.e. procrastinating).

You have a total of T hours that you can split among different subjects. For each subject i, the expected grade with t hours of studying is given by the function $f_i(t) = a_i t^2 + b_i t + c_i$, satisfying the following properties:

- $f_i(0) \geq 0$;
- $f_i(T) \leq 100$;
- $a_i < 0$;
- $f_i(t)$ is a non-decreasing function in the interval $[0, T]$.

You may allocate any fraction of an hour to a subject, not just whole hours. What is the maximum average grade you can obtain over all n subjects?

Input

The first line of each input contains the integers N $(1 \leq 10)$ and T $(1 \leq 240)$ separated by a space. This is followed by N lines, each containing the three parameters a_i, b_i, and c_i describing the function $f_i(t)$. The three parameters are separated by a space, and are given as real numbers with 4 decimal places. Their absolute values are no more than 100.

Output

Output in a single line the maximum average grade you can obtain. Answers within 0.01 of the correct answer will be accepted.

Sample Input	Sample Output
2 96 -0.0080 1.5417 25.0000 -0.0080 1.5417 25.0000	80.5696000000

Sample Input	Sample Output
3 34 -0.0657 4.4706 23.0000 -0.0562 3.8235 34.0000 -0.0493 3.3529 42.0000	70.0731488027

Problem F
Bumper-To-Bumper Traffic
Problem ID: traffic
Time Limit: 4 seconds

It's the slow crawl of rush hour. At any point of time, each vehicle is either stopped or is moving at the extremely slow speed of 1 meter per second. Lately, vehicles come equipped with a simple "black box" that record all changes in a vehicle's speed. In this problem, speeds change instantaneously.


Photo by mckay savage cc by-sa 2.0

The road is modelled as the real line (units in meters). So a car is identified with its position on the line. Also, cars are 4.4 meters long.

Given initial positions of two cars that are driving along the real line in the positive direction and a transcript of their speed changes, did these cars ever collide? While such a collision would be very slow speed (a "bumper tap"), any collision could result in erroneous readings from the black box in the future so the portions of the transcripts after a collision might not make sense.

Input

There is only one test case. The first line contains two integers $0 \leq X_1, X_2 \leq 10^6$ indicating the initial positions of the rears of the two vehicles in meters. You are guaranteed either $X_1 + 5 \leq X_2$ or $X_2 + 5 \leq X_1$. Initially (at time 0), the two cars are stopped.

The second line begins with a number $0 \leq N_1 \leq 10^5$ indicating the number of times the speed of the first car changed. The rest of the line contains N_1 integers $0 < T_1 < T_2 < \ldots < T_{n_1} \leq 10^6$ indicating the times (in seconds) the first vehicle changed speeds. So at time T_1 it begins driving at 1 m/s, at time T_2 it stops, at time T_3 it begins driving at 1 m/s, and so on.

The last line begins with a number $0 \leq N_2 \leq 10^5$ and is followed by N_2 integers $0 < T'_1 < T'_2 < \ldots < T'_{n_2} \leq 10^6$ that describe the times the second vehicle starts and stops.

Output

If the vehicles collide, output the message `bumper tap at time` S on a single line where S is the number of seconds from time 0 that the vehicles first collide, rounded up to the nearest second. If the vehicles do not collide, output the message `safe and sound` on a single line.

Sample Input	Sample Output

`0 5`
`3 1 4 5`
`3 1 4 6`
	`bumper tap at time 6`

Sample Input	Sample Output

`10 0`
`2 1 2`
`1 1`
	`bumper tap at time 8`

Sample Input	Sample Output

`2 13`
`1 1`
`3 4 7 10`
	`safe and sound`

Problem G
Flow Shop
Problem ID: flowshop
Time Limit: 6 seconds

Photo by ken figlioli cc by-sa 2.0

Sean's Swathers makes custom swathers (equipment used to harvest grain). All swathers go through the same basic stages in their construction: for example they all need to have a cutting bar, a grain belt, and a reel fitted. However, these components can be customized based on the buyer's needs, so these various stages may take different amounts of time between different swathers.

N swathers have been ordered and there are M stages in the manufacturing process. The swathers will each go through the same sequence of stages.

In particular, the processing occurs as follows: For each swather i and each stage j, it takes $P_{i,j}$ units of time to complete stage j for swather i. The workers at each stage may only work on one swather at a time. At the start of the day all swather orders are ready to be processed by the first stage. At any point in the process, if the workers at stage j are idle and there are swathers waiting to be processed at this stage then the workers will pick the swather that has the lowest label (they are labelled from 1 to N). Note that the work on a stage j can only be started after the work on the stage $j - 1$ is completed.

Determine the time each swather is completed.

Input

There is only one test case in each file. It begins with a single line containing N and M ($1 \leq N, M \leq 1000$), the number of swathers and stages (respectively). Following this are N lines, each with M integers. The j'th integer of the i'th line is $P_{i,j}$, giving the amount of time it will take for the workers at stage j to complete swather i ($1 \leq P_{i,j} \leq 10^6$).

Output

Output a single line containing N integers $T_1\ T_2\ \ldots\ T_n$ with a single space between consecutive integers. These should be such that stage M for swather i is completed at time T_i.

Sample Input	Sample Output
2 3 1 2 3 3 2 1	6 7

Sample Input	Sample Output
3 2 3 1 4 7 2 5	4 14 19

Problem H
Nice Numbers
Problem ID: numbers
Time Limit: 1 second

Image from wikipedia commons cc by-sa 3.0

If you are familiar with the game 2048, this problem may make sense right away.

Either way, we will define our own one-dimensional version of the game:

You are given a list of numbers containing only powers of 2. You can "compress" this list by "pushing" it to the right. If two identical numbers are next to each other, push will cause them to merge. The "merge" in this context means that the two numbers are replaced by their sum. Each number can be merged only once—if they can merge with either of their neighbors, they merge with the one to the right. This process is evaluated from the right. For example, the list $[2, 2, 2, 2]$ will become $[4, 4]$ after one push. For another example, given a list $[2, 2, 2]$, after a push, we end up with a new list $[2, 4]$ and we cannot change it further by "pushing".

Now, some of these lists, after some number of pushes, may end up with a single element. Using the example: $[8, 2, 2, 4] \Rightarrow [8, 4, 4] \Rightarrow [8, 8] \Rightarrow [16]$.

If the list can be reduced to a single element list only by "pushing", we call such a list **nice**.

Your task is to take a given list and make it **nice** by adding some (maybe zero) elements from $\{2, 4, 8\}$. To make this problem a bit easier, the initial list can contain only numbers in the set $\{2, 4, 8\}$.

Input

The first line of input contains a single positive integer $T \leq 100$ indicating the number of test cases. Next T lines each contains a string of length $1 \leq L \leq 100$, composed entirely of digits from the set $\{2, 4, 8\}$ (our representation of the given list).

Output

For each test case, output a line containing the shortest **nice** list built out of the input list by inserting zero or more of digits from the set $\{2, 4, 8\}$. If there are multiple optimal solutions, output the lexicographically smallest one.

Sample Input	Sample Output
3 222 8224 42424	2222 8224 422422448

This page is intentionally left blank.

Problem I
Stacking Cups
Problem ID: cups
Time Limit: 1 second

Photo by amy selleck cc by 2.0

You are programming a cup stacking module for your robot. This robot is equiped with several sensors that can accurately determine the radius and color of a cup. The problem is that there is a glitch in the robot's core routine that processess sensor inputs so the radius is doubled, if the result of the color sensor arrives to the routine after the radius.

For instance, for a red cup with a radius of 5 units, your module will receive either "red 5" or "10 red" message.

Given a list of messages from the core routine, each describing a different cup, can you put the cups in order of the smallest to the largest?

Input

The first line of the input file contains an integer N, the number of cups ($1 \leq N \leq 20$). Next N lines will contain two tokens each, either as "color radius" or "diameter color". The radius of a cup R will be a positive integer less than 1000. The color of a cup C will be a non-empty string of lower case English letters of length at most 20. All cups will be different in both size and color.

Output

Output colors of cups, one color per line, in order of increasing radius.

Sample Input	Sample Output
3 red 10 10 blue green 7	blue green red

This page is intentionally left blank.

Problem J
Stack Construction
Problem ID: stack
Time Limit: 2 seconds

You were recently hired by Rural and Municipal Roadway Communications to manage messages on a scrolling display above a major highway. Much to your surprise, these are very primitive displays. You have to input the message manually every time it should be changed (there is no memory to preload a list of messages).

Photo by petr katrochvil cc0 1.0

Strangely, the only way to post messages is using an on-board stack. You can push a character onto the top of the stack, you can pop the character that is on top of the stack, and you can print the character that is on top of the stack..

Out of boredom, or perhaps the universal human desire to do as little work as possible to get the job done, you wonder what the minimum number of `push`, `pop`, and `print` are required to print a message your boss has told you to display. Oh, you must also ensure the stack is clear at the end so that you are ready to input a new message next time your boss asks you to do this.

Example If we want to print the message `abba` and then clear the stack you could do the following. Note the contents of the stack are recorded below with the top of the stack on the right.

	operation	**stack contents**	**displayed message**
1	`push a`	`a`	
2	`print`	`a`	`a`
3	`push b`	`ab`	`a`
4	`print`	`ab`	`ab`
5	`print`	`ab`	`abb`
6	`pop`	`a`	`abb`
7	`print`	`a`	`abba`
8	`pop`		`abba`

In fact, this is the fewest operations that can be performed to print exactly the message `abba` and leave the stack empty.

Input

The first line of input is a single integer $T \leq 30$ indicating the number of test cases. Each of the following T lines contains a single string consisting of any printable characters. The first and last character of each line will not be a space. Each line has at least one and at most 200 characters.

Output

For each of the T strings in the input, you should output on a single line the minimum number of operations required to print the string on the display.

Sample Input	Sample Output
4	3
d	8
abba	34
rollover ahead	38
ogopogo spotted!	